IFW AF

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Applicant:	§	
	§	
MOSHE GEFEN ET AL.	§	
	§	
Serial No.: 09/629,966	§	
	§	
Filed: July 31, 2000	§	Group Art Unit: 2186
	§	
For: SYSTEM AND METHOD FOR ENABLING NON-VOLATILE MEMORY TO EXECUTE CODE WHILE OPERATING AS A DATA STORAGE/PROCESSING DEVICE	§	Attorney Docket: 246/68
	§	
Examiner: Matthew D. Anderson	§	

TRANSMITTAL OF APPEAL BRIEF

Commissioner of Patents and Trademarks
Washington, DC 20231

Dear Sir:

Responsive to a notification of noncompliance with 37 CFR 1.192(c) mailed September 30, 2004 (copy enclosed), submitted herein is a replacement Appeal Brief in triplicate. The Appeal Brief fees were earlier submitted with the original Appeal Briefs. In the event that the Appeal Brief fees have not been charged, authorization is hereby granted to charge the appropriate fees, belived to be $340, to account no. 06-2140. A duplicate copy of this transmittal letter is attached.

Respectfully submitted,

Mark M. Friedman
Attorney for Applicant
Registration No. 33,883

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Applicant:	§	
	§	
MOSHE GEFEN ET AL.	§	
	§	
Serial No.: 09/629,966	§	
	§	
Filed: July 31, 2000	§	Group Art Unit: 2186
	§	
For: SYSTEM AND METHOD FOR ENABLING NON-VOLATILE MEMORY TO EXECUTE CODE WHILE OPERATING AS A DATA STORAGE/PROCESSING DEVICE	§	Attorney Docket: 246/68
	§	
Examiner: Matthew D. Anderson	§	

Commissioner of Patents and Trademarks
Washington, DC 20231
ATTENTION: Board of Patent Appeals and Interferences

APPELLANT'S BRIEF

Dear Sir:

This is in furtherance of the Notice of Appeal filed in this case on August 30, 2004.

The fees required under § 1.17(f) and any required petition for extension of time for filing this brief and fees therefor are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief is transmitted in triplicate.

This brief contains these items under the following headings and in the order set forth below:

I. REAL PARTY IN INTEREST

II. RELATED APPEALS AND INTERFERENCES

III. STATUS OF CLAIMS

IV. STATUS OF AMENDMENTS

V. SUMMARY OF CLAIMED SUBJECT MATTER

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

VII. ARGUMENTS

_X_ ARGUMENT: VIIIA REJECTIONS UNDER 35 U.S.C. 112, FIRST PARAGRAPH

_X_ ARGUMENT: VIIIC REJECTIONS UNDER 35 U.S.C. 102

_X_ ARGUMENT: VIIID REJECTIONS UNDER 35 U.S.C. 103

VIII. APPENDIX OF CLAIMS INVOLVED IN THE APPEAL

IX APPENDIX OF EVIDENCE

X. APPENDIX OF RELATED PROCEEDINGS

I. REAL PARTY IN INTEREST

The real party in interest in this case is:

M-Systems Flash Disk Pioneers, Ltd.

Central Park 2000

Atir Yeda 7

44425 Kfar Saba

Israel

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals and interferences.

III. STATUS OF CLAIMS

Claims 1, 3-6, 10, 13, 14 and 16-30 are being appealed. There are no allowed claims.

IV. STATUS OF AMENDMENTS

In response to the Official Action mailed on January 13, 2004, independent claims 13, 14, 16 and 17 and dependent claims 22 and 23 were amended, and new dependent claims 29 and 30 were added.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 recites a system that uses a non-volatile memory device both for executing code and for data processing operations. The system comprises a host for accessing the non-volatile memory device (page 7 lines 10-11; Figure 4 item 20), a non-volatile array for holding the code and the data (claim 1 as filed; Figure 4 item 24), non-volatile device circuitry for controlling the non-volatile array (claim 1 as filed; Figure 4 item 25) and a logic circuit for enabling automatic suspending (page 8 line 14 through page 9 line 14; Figure 4 item 26) and/or automatic resuming (page 9 line 15 through page 10 line 5; Figure 4 item 27) of operations. The logic circuit is separate from the host (Figure 4). The suspending and resuming of operations is in response to a read request from the host (page 9 lines 1-2).

Independent claim 13 recites a method of executing code while processing data on a non-volatile memory device (Figure 4 items 23, 24 and 25). At least one logic circuit (page 4 line 20; Figure 4 items 26 and 27) is added to the device. The logic circuit(s) monitor(s) the status of current operations in the device (page 8 lines 16-20), signal(s) if the device is available for code execution (page 9 lines 10-14) and command(s) the device to suspend and/or resume device operations in response to a read request (page 9 lines 1-2; page 9 lines 16-17; page 9 line 21 through page 10 line 1).

Independent claim 14 also recites a method of executing code while processing data on a non-volatile memory device (Figure 4 items 23, 24 and 25). At least one logic circuit (page 4 line 20; Figure 4 items 26 and 27) is added to the device. The logic circuit(s) sense(s) a read request while the device is in program/erase mode/operation (page 8 lines 16-20; a read operation is called a "read request" on page 6 line 19; Figure 3 item 11). In response to sensing a read request,

the logic circuit(s) enter the program/erase operations into suspended mode (page 9 lines 1-2; Figure 3 item 12), signal(s) to the CPU/Bus to delay executing the read request (page 9 lines 7-10), turn(s) off the signal to allow the CPU/Bus to execute the read request (page 9 lines 11-14) and exits the device form the suspended mode to continue the program/erase operation (page 9 line 16 through page 10 line 1).

Independent claim 16 recites a single flash memory device that comprises two logic circuits: a suspend logic circuit for enabling hardware-initiated suspending of data processing operations (page 8 line 14 through page 9 line 14; claim 16 as filed; Figure 4 item 26) in response to read requests received from a host (page 9 lines 1-2; page 6 line 22 read in conjunction with page 7 lines 10-11; Figure 4 item 20) and a resume logic circuit for enabling hardware-initiated resuming of data processing operations (page 9 line 15 through page 10 line 5; claim 16 as filed; Figure 4 item 27).

Independent claim 17 recites a memory device that comprises a non-volatile memory (Figure 4 item 24); circuitry for reading, programming and erasing the non-volatile memory (Figure 4 item 25); and a hardware mechanism for suspending circuitry activity in response to read requests (page 9 lines 1-2; Figure 4 item 26) received from a host (page 6 line 22 read in conjunction with page 7 lines 10-11; Figure 4 item 20).

Independent claim 24 recites a method for managing a memory device that includes a non-volatile memory (Figure 4 item 24) and that is accessed by a host (page 7 lines 10-11; Figure 4 item 20). The device commences either an erase operation of the memory or a programming operation of the memory (page 6 line 18; page 7 line 5; page 8 lines 16-17; Figure 3 item 15). During the operation, the host requests a read operation (page 6 lines 18-19; Figure 3 item 11). In response to the

read request, the device suspends the erase/programming operation (page 9 lines 1-2; Figure 3 item 12).

Dependent claim 25 adds to claim 24 that in response to the read request, the memory device signals the host to delay execution of the read request (page 9 lines 7-10). Dependent claim 26 adds to claim 25 that the host responds to the signal by delaying execution of the read request (page 7 lines 10-13; page 9 lines 8-10). Dependent claim 27 adds to claim 26 that the memory device signals the host to resume execution of the read request (page 9 lines 11-14). Dependent claim 28 adds that after suspending the erase/programming operation, the memory device monitors a conclusion of read requests from the host (page 6 line 22 through page 7 line 1; page 9 lines 18-20; Figure 3 item 13) and, upon detecting that conclusion, resumes the erase/programming operation (page 7 lines 1-2; page 9 line 20 through page 10 line 1; Figure 3 item 14).

Dependent claim 29 adds to claim 24 the limitation that the suspending of the erase/programming operation is effected only by the memory device (page 8 line 14 through page 9 line 14; Figure 4 item 26). Dependent claim 30 adds to claim 28 the limitation that the resuming of the erase/programming operation is effected only by the memory device (page 9 line 15 through page 10 line 5; Figure 4 item 27).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The issues presented for review are:

1. Whether claims 29 and 30 contain subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the art that the inventors, at the time the application was filed, had possession of the claimed invention;

2. Whether claims 1, 3, 4, 6, 10, 13, 14 and 16-30 are anticipated by See et al., US Patent No. 6,189,070 (henceforth, "See et al. '070"); and

3. Whether claim 5 is unpatentable over See et al. '070 and Keeley et al., US Patent No. 4,491,790 (henceforth, "Keeley et al. '790").

VII. ARGUMENTS

REJECTIONS UNDER 35 U.S.C. 112, FIRST PARAGRAPH

The Examiner rejected claims 29 and 30 under 35 U.S.C. 112, first paragraph, as containing subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventors, at the time the application was filed, had possession of the claimed invention. Specifically, the Examiner found no support in the specification for the suspending/resuming being effected only by the memory device.

As noted by the Examiner, newly added claim limitations must be supported in the specification through explicit, implicit or inherent disclosure. In the present case, the limitation in question is supported at least implicitly, if not explicitly, in the specification. Specifically, the specification describes how the suspending and resumption of a program/erase operation is managed exclusively by automatic suspend logic **16** and automatic resume logic **27** that are illustrated in Figure 4 as components of the memory device, within the rectangle that encloses the other components (Bus I/F logic **23**, Flash array **24** and Flash circuitry **25**) of the memory device. Page 8 line 14 through page 9 line 14 describes how the entry to the suspend state is managed exclusively by automatic suspend logic **26**. Page 9 line 15 through page 10 line 5 of the specification describes how the resumption of the suspended operation is managed exclusively by automatic resume logic **27**.

Consider, first the description of the entry to the suspend state. Entry to the suspend state is contingent on the host attempting to read from the memory device. As stated on page 9 lines 1-13,

> Upon detection of the read operation **11** the automatic suspend logic **26** executes a process that enters the device into the suspend state **12**...In addition, the logic may mark in a certain place (e.g. I/O port or a dedicated register) that the device has entered the automatic suspend

> state **12**...In addition, the logic will indicate that the device is on its way to the automatic suspend state **12** using an external signal (Busy signal) **22**...The logic is also responsible of verifying that the device has actually entered the automatic suspend state. After the verification phase – the Busy signal will be turned off (to indicate that the device has entered the automatic suspend state **12**. (emphasis added)

Note in particular that all the relevant operations (entering the suspend state, marking the entry to the suspend state, turning the Busy signal on and off) are performed only by automatic suspend logic **26**.

Now consider the resumption of the suspended operation, as described on page 9 line 16 through page 10 line 5:

> The automatic resume logic **27** starts to operate when the device enters the automatic suspend state **12**. The target of this logic is to resume the program/erase operation **15** that was interrupted by the automatic suspend logic **26**. This logic should monitor the read operations done from the device, for example, by using the same techniques as the automatic suspend logic **26**. The logic is responsible to resume the suspended operation. One suggested implementation is to wait for a break in the read operations of the device. When the break is long enough...the logic executes a process which causes the device to resume the program/erase operation **15**...The logic contains some mechanism to determine if the break is a real break or just a temporary break...The logic is also responsible to turn off the mark that shows...that the device has entered the automatic suspend state **12**. (emphasis added)

Again, all the relevant operations (monitoring for a break in the read operations, distinguishing a real break from a temporary break, unmarking the entry to the suspended state, resuming the program/erase operation) are performed only by automatic resume logic **27**.

REJECTIONS UNDER 35 U.S.C. 102

The Examiner rejected claims 1, 3, 4, 6, 10, 13, 14 and 16-30 as being anticipated by See et al. '070.

See et al. '070 teach a memory device **410** whose programming/erasing operations are suspended when processor **402** that accesses device **410** needs to read data from device **410**. (Note that the processor in Figure 2 is labeled "**400**". This is an error. The reference numeral that is used consistently in the text of See et al. '070 for the processor is "**402**".) The difference between the present invention and the invention of See et al. '070 lies in which component is responsible for initiating the suspensions of programming/erasing operations. In See et al. '070 processor **402** is responsible for signaling device **410** explicitly to initiate suspension and resumption of the programming/erasing operations as needed. This is in contrast to the present invention, in which hardware in the memory device itself initiates the suspension and resumption of the programming/erasing operations in response to read requests from the host system.

It is clear from See et al. '070 that it is their processor **402**, and not the hardware in signaling device **410** such as circuitry **190**, **192**, **194** and **195**, that takes the initiative in suspending and resuming programming/erasing operations. This aspect of the invention of See et al. '070 is clearest in their claims, for example claim 6:

> A system comprising:
> a processor;
> a nonvolatile writeable memory having an array, the nonvolatile writeable memory having a first mode for allowing the array to be written to, and a second mode for allowing the array to be read from, the nonvolatile writeable memory storing low level code which when executed by the processor cause the processor to perform:
> (a) disabling interrupts;

(b) initiating a non-read operation to the nonvolatile writeable memory while the array is in the first mode and, wherein, the non-read operation is initiated by low-level routines downloaded to a volatile memory from the nonvolatile writeable memory;
(c) checking for interrupts, and in response to detecting an interrupt performing:
(i) suspending the non-read operation;
(ii) placing the array in the second mode;
(iii) enabling interrupts; and
(iv) reading code from the nonvolatile writeable memory. (emphasis added)

and 11:

A computer-readable medium having stored thereon a plurality of instructions which, when executed by a processor, cause the processor to perform:
(a) disabling interrupts
(b) initiating a non-read operation to the nonvolatile writeable memory while the array is in a first mode and, wherein, the non-read operation is initiated by low-level routines downloaded to a volatile memory from the nonvolatile writeable memory;
(c) checking for interrupts, and in response to detecting an interrupt performing the steps of:
(i) suspending the non-read operation;
(ii) placing the array in a second mode;
(iii) enabling interrupts; and
(iv) reading code from the nonvolatile writeable memory. (emphasis added)

This aspect of the invention of See et al. '070 also is clear from column 8 lines 65-67:

The program suspend operation is initiated by writing a program suspend command to the command decoder **170**.

and from column 9 lines 5-7:

The erase suspend operation may be initiated by writing an erase suspend command to the command decoder **170**.

Note in particular that from among the commands decoded by command decoder **170**,

as listed in column 7 lines 50-52:

...(1) erase, (2) erase suspend, (3) erase resume, (4) program, (5) program suspend, (6) program resume, (7) read, and (8) read status

the "read" command is specifically not used to initiate the suspending and resuming of programming/erasing operations.

This distinction between the present invention and See et al. '070 is recited explicitly in independent claims 1, 13, 14, 16, 17 and 24.

The last element recited in claim 1 is:

> logic circuit, separate from said host, for enabling automatic suspending and/or automatic resuming of operations in response to a read request from said host (emphasis added)

The first step recited in claim 13 is:

> adding at least one logic circuit to the non-volatile memory device

The last step recited in claim 13 is:

> commanding the device to suspend and/or resume device operations in response to a read request, by said at least one logic circuit (emphasis added)

The first three steps recited in claim 14 is:

> adding at least one logic circuit to the non-volatile memory device
>
> sensing a read request while the device is in program/erase mode/operation, by said at least one logic circuit (emphasis added)

and

> in response to said sensing, entering of program and/or erase operations into suspended mode, by said at least one logic circuit (emphasis added)

The last step recited in claim 14 is

> exiting of said device from said suspended mode to continue program/erase operation, by said at least one logic circuit (emphasis added)

Claim 16 recites a flash memory device comprising two elements:

> a suspend logic circuit for enabling hardware initiated suspending of data processing operations in response to at least one read request received by the memory device from a host (emphasis added)

and

> a resume logic circuit for enabling hardware initiated resuming of data processing operations

Claim 17 recites a memory device whose third element is

> a hardware mechanism for suspending an activity of said circuitry in response to at least one read request received by the memory device from a host (emphasis added)

The last two steps recited in claim 24 are:

> during said operation, requesting a read operation, by the host (emphasis added)

and

> in response to said request, suspending said operation, by the memory device (emphasis added)

These arguments were presented in response to the Official Action mailed January 13, 2004. In response to these arguments, the Examiner noted that

> The only circuitry explicitly disclosed in the See involving the suspension of programming/erasing is shown in the erase/program suspend circuitry (items 192 & 195 in figure 7) and erase and program suspend and resume latches (178 b, c, e, f).

The Examiner's observation is correct but irrelevant. Erase suspend circuitry **192**, program suspend circuitry **195**, erase latch **176a**, erase suspend latch **176b**, program latch **176d** and program suspend latch **176e** are controlled by command decoder **170** that in turn is controlled by commands received from processor **402**. Among the commands that command decoder **170** receives from processor **402** are (column 7 lines 50-51):

> (2) erase suspend, (3) erase resume...(5) program suspend, (6) program resume

So even though the only hardware recited explicitly by See et al. '070 for suspending and resuming programming and erasing operations is inside flash device **410**, it is

clear that processor **402** participates actively in the suspending and resuming programming and erasing operations.

Furthermore, the Examiner did not even address the second aspect of the present invention, as recited in independent claims 1, 13, 14, 16, 17 and 24, that distinguishes the present invention from the teachings of See et al. '070. Flash device **410** of See et al. '070 suspends an erase operation in response to an erase suspend command, not in response to a read command. Flash device **410** of See et al. '070 suspends a programming operation in response to a program suspend command, not in response to a read command. By contrast, the memory device of the present invention suspends erase and programming operations in response to read commands/requests.

Therefore, independent claims 1, 13, 14, 16, 17 and 24 are not anticipated by See et al. '070.

Furthermore, independent claims 1, 13, 14, 16, 17 and 24 are not obvious from See et al. '070. There is neither a hint nor a suggestion in See et al. '070 that flash device **410** should suspend programming operations or erase operations in response to a read command from processor **402**.

Claims 3, 4, 6, 10, 18-23 and 25-30 are allowable by virtue of depending from independent claims 1, 17 and 24.

REJECTIONS UNDER 35 U.S.C. 103

The Examiner rejected claim 5 as unpatentable over See et al. '070 and Keeley et al. '790.

It is demonstrated above that independent claim 1 is allowable. It follows that claim 5, that depends therefrom, also is allowable.

Respectfully submitted,

Mark M. Friedman
Attorney for Applicant
Registration No. 33,883

Date: November 23, 2004

VIII. APPENDIX OF CLAIMS INVOLVED IN THE APPEAL

The text of the claims on appeal is:

1. A system that executes code while processing data operations using a non-volatile memory device, comprising:

host for accessing said memory device;

non volatile array for holding code and data of said system;

non volatile device circuitry for controlling content and activity of said non volatile array; and

logic circuit, separate from said host, for enabling automatic suspending and/or automatic resuming of operations in response to a read request from said host.

3. The system of claim 1, wherein said non-volatile memory device is a flash memory device.

4. The system of claim 1, wherein said logic circuit enables code execution and data storage/processing facilities within a single chip device with a single silicon die.

5. The system of claim 1, wherein said logic circuit enables code execution and data storage/processing facilities within a bank of single memory chips with single silicon dies.

6. The system of claim 1, wherein said logic circuit is embedded into the memory device.

10. The system of claim 1, wherein said logic circuit is operative to monitor status of current operations in said memory device.

13. A method for executing code while processing data on a non-volatile memory device, comprising the steps of:

i. adding at least one logic circuit to the non-volatile memory device;

ii. monitoring status of current operations in said memory device, by said at least one logic circuit;

iii. signaling if the device is available for code execution, by said at least one logic circuit; and

iv. commanding the device to suspend and/or resume device operations in response to a read request, by said at least one logic circuit.

14. A method for executing code while processing data on a non-volatile memory device, comprising the following steps:

i. adding at least one logic circuit to the non-volatile memory device;

ii. sensing a read request while the device is in program/erase mode/operation, by said at least one logic circuit;

iii. in response to said sensing, entering of program and/or erase operations into suspended mode, by said at least one logic circuit;

iv. signaling to CPU/Bus to delay executing said read request, by said at least one logic circuit;

v. turning off signal to allow CPU/Bus to execute said read request, by said at least one logic circuit; and

vi. exiting of said device from said suspended mode to continue program/erase operation, by said at least one logic circuit.

16. A single flash memory device comprising:

a suspend logic circuit for enabling hardware initiated suspending of data processing operations in response to at least one read request received by the memory device from a host; and

a resume logic circuit for enabling hardware initiated resuming of data processing operations.

17. A memory device comprising:

(a) a non-volatile memory;

(b) circuitry for reading, programming and erasing said non-volatile memory; and

(c) a hardware mechanism for suspending an activity of said circuitry in response to at least one read request received by the memory device from a host.

18. The memory device of claim 17, wherein said hardware mechanism also is operative to resume said activity of said circuitry after said circuitry has finished processing said at least one read request.

19. The memory device of claim 17, wherein said activity is erasing said non-volatile memory.

20. The memory device of claim 17, wherein said activity is programming said non-volatile memory.

21. The memory device of claim 17, wherein said hardware mechanism includes at least one logic circuit.

22. The memory device of claim 17, wherein said suspending of said activity includes:

(A) indicating to said host that issued said at least one read request that execution of said at least one read request should be delayed; and

(B) subsequently, indicating to said host that the memory device is available for reading.

23. The memory device of claim 17, wherein said hardware mechanism is further operative to monitor said processing of said at least one read request to determine when said circuitry has finished processing said at least one read request.

24. A method for managing a memory device that includes a non-volatile memory and that is accessed by a host, comprising the steps of:

(a) commencing an operation selected from the group consisting of erasing the non-volatile memory and programming the non-volatile memory, by the memory device;

(b) during said operation, requesting a read operation, by the host; and

(c) in response to said request, suspending said operation, by the memory device.

25. The method of claim 24, further comprising the step of:

(d) in response to said request, signaling to the host to delay execution of said request, by the memory device.

26. The method of claim 25, further comprising the step of:

(e) in response to said signal, delaying execution of said request, by the host.

27. The method of claim 26, further comprising the step of:

(f) signaling the host to resume execution of said request, by the memory device.

28. The method of claim 27, further comprising the steps of:

(g) subsequent to said suspending, monitoring a conclusion of read requests from the host, by the memory device; and

(h) upon detecting said conclusion, resuming said operation, by the memory device.

29. The method of claim 24, wherein said suspending is effected only by the memory device.

30. The method of claim 28, wherein said resuming is effected only by the memory device.

IX. APPENDIX OF EVIDENCE

NONE

X. APPENDIX OF RELATED PROCEEDINGS

NONE